

# 2단계 파이프라인구조의 64B/66B 인코더/디코더를 이용한 물리적 선로 부계층 설계

## Design of PCS with two stage pipelining 64B/66B Encoder/Decoder

Jin-Cheol Song\*, Tae-Ho Kim\*\*, Jin-Ku Kang\*\*\*  
송진철\*, 김태호\*\*, 강진구\*\*\*

### Abstract

In this paper, to implement PCS (Physical Coding Sublayer) of 10GBASE-R type, we present 2 stage pipeline 64b/66b Encoder/Decoder which operates at 156.25MHz standard specification and designed to minimize clock latency as possible as we can. The proposed circuit was designed based on Verilog hardware description language and measured for functional verification on VertexII-1000fg456 chip of Xilinx Inc.. Total equivalent gate count is 47,303 and estimated power consumption is 351mW at Vcc 3.3V.

### 요약

본 논문에서는 10GBASE-R 형식의 PCS (Physical Coding Sublayer) 구현을 위한 회로로써 표준 속도인 156.25MHz에서 동작하면서 2단 파이프라인 구조로 64b/66b 인코더/디코더를 설계하여 가능한 클럭 지연을 최소화한 회로를 제시한다. 제안하는 PCS 회로는 Verilog 하드웨어 설계 언어를 기반으로 설계하여 FPGA를 통한 기능검증을 위해 Xilinx사의 VertexII-1000fg456 칩에서 측정하였다. 측정된 게이트 수는 47,303이고, Vcc 3.3V에서 351mW의 전력 소모를 보였다.

Key words : PCS, Pipeline, 64B/66B, Encoder/Decoder, Scrambler

### 1. 서론

10기가 이더넷 물리계층의 구조는 그림 1과 같이 이에 대한 표준의 하나로서 LAN 방식에 쓰이는 R형식의 PCS(Physical Coding Sublayer- 물리적 선로 부계층)는 MAC(Media Access Controller) 프레임으로부터 받는 데이터를 인코딩하기 위해 이전 세대의 이더넷에서 쓰던 8B/10B 전송코드 대신 64B/66B 전송코드 방식을 사용하여 코딩 시 발생하는 오버헤드를 기존의 25%에서 3%로 줄일 수 있게 되는 이점을 가진다[1]. 위와 같은 변화는 데이터 전송대역이 10기가인 WAN 백본의 OC-192의 속도와의 매치가 되어

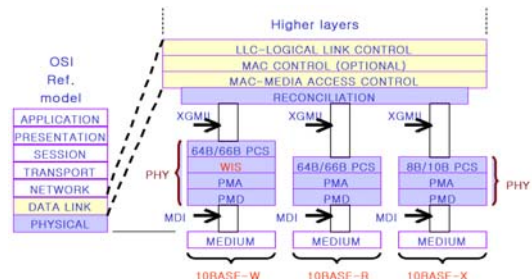


Fig. 1. Architecture of 10Gigabit Ethernet

그림 1. 10기가비트 이더넷의 계층구조

SONET/SDH와 동급의 효과를 지니면서도 가격은 1/10으로 구현이 되기에 경제적 측면에서도 이점을 볼 수 있는 장점을 가진다. 이로 인하여 백본망에 SONET framer 추가가 가능하게 되었고 W 형식으로 물리층이 표준화가 되어 LAN/MAN/WAN을 하나의 네트워크로 통합하는 것이 가능하게 되었다[2].

본 논문에서는 IEEE802.3ae 그룹에서 표준화된 10기가비트 이더넷 백본망의 물리적 선로부호 중 R형식의 PCS를 제안하려 한다. 본 논문에서의 PCS는 표준속도인 156.25MHz에서 동작하면서 클럭 지연을 줄이고자 2단 파이프라인 구조의 64b/66b 인코더와 디코더를 사용하였고 인코딩된 데이터의 스크램블을 위해 병렬 스크램블러를 설계하였다.

또한 SERDES로 데이터 전송을 위해 데이터 크기

\* (주) 실리콘웍스 (Silicon Works Co., Ltd.)

\*\* 인하대학교 전자공학부  
(School of Electronics Engineering, Inha University)

★ 교신저자 (Corresponding author)

※ 감사의 글 (Acknowledgment)

본 연구는 인하대학교 교내 연구비 지원으로 수행되었음.  
반도체설계교육센터(IDEC)에서 제공된 설계 툴을 지원받아 수행되었음.

接受日:2009年 12月 6日, 修正完了日: 2009年 12月 23日

를 조정해주고 동기를 맞추는 기어박스과 동기화회로를 설계하였다. 이후의 절에서는 실질적인 PCS가 포함되는 물리층에 대해 설명하고 64b/66b 전송코드와 스크램블러의 다항식에 대해 설명하고 구현한 회로의 실험결과를 서술할 것이다.

### II. 구조

그림 2는 PCS의 구조와 이더넷 백본에서 PCS와 인접한 인터페이스를 도시화한 것이다. MAC(Media Access Controller) 층에서 나온 데이터를 PCS를 통해 채널로 보내주는 방향을 송신 방향으로 하고 역방향인 채널에서 MAC으로 가는 것을 수신 방향으로 정의한다.

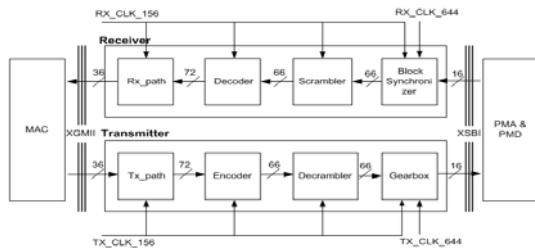


Fig. 2. Block diagram of 10GBASE-R PCS and adjacent interfaces

그림 2. 10GBASE-R PCS 블록의 구성과 인접 인터페이스

PCS와 MAC 사이의 인터페이스를 XGMII (10 Gigabit Media Independent Interface)라 하고 여기서의 데이터 대역은 156.25MHz 속도에서 10Gb/s의 데이터 대역을 갖게 된다. PCS와 PMA간의 인터페이스를 XSBI(10 Gigabit Sixteen Bit Interface)라고 한다. 인터페이스의 데이터 라인은 16비트의 병렬 데이터인데 이는 직병렬화기의 데이터 라인 수를 2n 형태로 변환하기 위해서이다. 따라서 표준 규격을 따를 시 644.53MHz에서 10.3126Gb/s가 된다. PCS 내부에서는 인코더/디코더와 스크램블러/디스크램블러가 데이터의 부호화와 복원을 담당하는 회로이다. 인코더/디코더는 64B/66B 전송코드에 따라서 그리고 스크램블러/디스크램블러는  $G(x) = 1 + x^{39} + x^{58}$ 의 다항식에 따라 데이터를 부호화 및 복원을 수행한다[3],[4].

### III 회로 설계

제한하는 PCS는 기본적으로 블록데이터의 인코딩과 디코딩을 수행하는 기능을 갖는 송수신기 회로이다. 기본 구조에 대한 블록 다이어그램을 그림 2에 보였다. 전방향 통신방식인 R 형식의 10기가 PCS는 그림 2와 같이 송수신단이 분리되어있다. 데이터 송수신시 각 기능을 담당하는 회로들이 순차적으로 데이터를 가공한다. 인코더와 디코더 회로에 대한 설명은 다음과 같다.

표 1과 표 2는 64B/66B 인코더/디코더의 부호화가 어떠한 형태와 값으로 이루어지는지 나타내는 블록 포맷과 전송 코드 암호표를 나타낸다. 전송코드에 의

해 인코딩된 66비트 그룹의 코드는 다음과 같이 크게 두 가지의 형태로 나눌 수 있다. 첫 번째는 입력받은 8레인이 모두 데이터 기호일 경우와 두 번째는 그렇지 않을 경우 즉, 모든 레인이 컨트롤 기호이거나 데이터와 컨트롤이 혼합된 경우이다.

Table 1. Block format of 64B/66B

표 1. 64B/66B 블록 포맷

Input Data	S y n c	Block Payload													
BE Position: 0 1 2		65													
Data Block Format:	D <sub>0</sub> D <sub>1</sub> D <sub>2</sub> D <sub>3</sub> D <sub>4</sub> D <sub>5</sub> D <sub>6</sub> D <sub>7</sub>	D <sub>0</sub>	D <sub>1</sub>	D <sub>2</sub>	D <sub>3</sub>	D <sub>4</sub>	D <sub>5</sub>	D <sub>6</sub>	D <sub>7</sub>						
Control Block Formats:	Block Type Field														
C <sub>0</sub> C <sub>1</sub> C <sub>2</sub> C <sub>3</sub> C <sub>4</sub> C <sub>5</sub> C <sub>6</sub> C <sub>7</sub>	0x1e	C <sub>0</sub>	C <sub>1</sub>	C <sub>2</sub>	C <sub>3</sub>	C <sub>4</sub>	C <sub>5</sub>	C <sub>6</sub>	C <sub>7</sub>						
C <sub>0</sub> C <sub>1</sub> C <sub>2</sub> C <sub>3</sub> D <sub>4</sub> D <sub>5</sub> D <sub>6</sub> D <sub>7</sub>	0x2f	C <sub>0</sub>	C <sub>1</sub>	C <sub>2</sub>	C <sub>3</sub>	D <sub>4</sub>	D <sub>5</sub>	D <sub>6</sub>	D <sub>7</sub>						
C <sub>0</sub> C <sub>1</sub> C <sub>2</sub> C <sub>3</sub> S <sub>4</sub> D <sub>5</sub> D <sub>6</sub> D <sub>7</sub>	0x33	C <sub>0</sub>	C <sub>1</sub>	C <sub>2</sub>	C <sub>3</sub>					D <sub>5</sub>	D <sub>6</sub>	D <sub>7</sub>			
O <sub>0</sub> D <sub>1</sub> D <sub>2</sub> D <sub>3</sub> S <sub>4</sub> D <sub>5</sub> D <sub>6</sub> D <sub>7</sub>	0x66	D <sub>1</sub>	D <sub>2</sub>	D <sub>3</sub>	O <sub>4</sub>	D <sub>5</sub>	D <sub>6</sub>	D <sub>7</sub>							
O <sub>0</sub> D <sub>1</sub> D <sub>2</sub> D <sub>3</sub> O <sub>4</sub> D <sub>5</sub> D <sub>6</sub> D <sub>7</sub>	0x55	D <sub>1</sub>	D <sub>2</sub>	D <sub>3</sub>	O <sub>4</sub>	D <sub>5</sub>	D <sub>6</sub>	D <sub>7</sub>							
S <sub>0</sub> D <sub>1</sub> D <sub>2</sub> D <sub>3</sub> D <sub>4</sub> D <sub>5</sub> D <sub>6</sub> D <sub>7</sub>	0x78	D <sub>1</sub>	D <sub>2</sub>	D <sub>3</sub>	D <sub>4</sub>	D <sub>5</sub>	D <sub>6</sub>	D <sub>7</sub>							
O <sub>0</sub> D <sub>1</sub> D <sub>2</sub> D <sub>3</sub> C <sub>4</sub> C <sub>5</sub> C <sub>6</sub> C <sub>7</sub>	0x4b	D <sub>1</sub>	D <sub>2</sub>	D <sub>3</sub>	O <sub>4</sub>	C <sub>5</sub>	C <sub>6</sub>	C <sub>7</sub>							
T <sub>0</sub> C <sub>1</sub> C <sub>2</sub> C <sub>3</sub> C <sub>4</sub> C <sub>5</sub> C <sub>6</sub> C <sub>7</sub>	0x87				C <sub>1</sub>	C <sub>2</sub>	C <sub>3</sub>	C <sub>4</sub>	C <sub>5</sub>	C <sub>6</sub>	C <sub>7</sub>				
D <sub>0</sub> T <sub>1</sub> C <sub>2</sub> C <sub>3</sub> C <sub>4</sub> C <sub>5</sub> C <sub>6</sub> C <sub>7</sub>	0x99	D <sub>0</sub>			C <sub>2</sub>	C <sub>3</sub>	C <sub>4</sub>	C <sub>5</sub>	C <sub>6</sub>	C <sub>7</sub>					
D <sub>0</sub> D <sub>1</sub> T <sub>2</sub> C <sub>3</sub> C <sub>4</sub> C <sub>5</sub> C <sub>6</sub> C <sub>7</sub>	0xaa	D <sub>0</sub>	D <sub>1</sub>			C <sub>3</sub>	C <sub>4</sub>	C <sub>5</sub>	C <sub>6</sub>	C <sub>7</sub>					
D <sub>0</sub> D <sub>1</sub> D <sub>2</sub> T <sub>3</sub> C <sub>4</sub> C <sub>5</sub> C <sub>6</sub> C <sub>7</sub>	0xb4	D <sub>0</sub>	D <sub>1</sub>	D <sub>2</sub>			C <sub>4</sub>	C <sub>5</sub>	C <sub>6</sub>	C <sub>7</sub>					
D <sub>0</sub> D <sub>1</sub> D <sub>2</sub> D <sub>3</sub> T <sub>4</sub> C <sub>5</sub> C <sub>6</sub> C <sub>7</sub>	0xcc	D <sub>0</sub>	D <sub>1</sub>	D <sub>2</sub>	D <sub>3</sub>			C <sub>5</sub>	C <sub>6</sub>	C <sub>7</sub>					
D <sub>0</sub> D <sub>1</sub> D <sub>2</sub> D <sub>3</sub> D <sub>4</sub> T <sub>5</sub> C <sub>6</sub> C <sub>7</sub>	0x62	D <sub>0</sub>	D <sub>1</sub>	D <sub>2</sub>	D <sub>3</sub>	D <sub>4</sub>		C <sub>6</sub>	C <sub>7</sub>						
D <sub>0</sub> D <sub>1</sub> D <sub>2</sub> D <sub>3</sub> D <sub>4</sub> D <sub>5</sub> T <sub>6</sub> C <sub>7</sub>	0xe1	D <sub>0</sub>	D <sub>1</sub>	D <sub>2</sub>	D <sub>3</sub>	D <sub>4</sub>	D <sub>5</sub>		C <sub>7</sub>						
D <sub>0</sub> D <sub>1</sub> D <sub>2</sub> D <sub>3</sub> D <sub>4</sub> D <sub>5</sub> D <sub>6</sub> T <sub>7</sub>	0xff	D <sub>0</sub>	D <sub>1</sub>	D <sub>2</sub>	D <sub>3</sub>	D <sub>4</sub>	D <sub>5</sub>	D <sub>6</sub>							

Table 2. Transmission encoding table  
표 2. 전송 코드 암호표

Control Character	8bit XGMII Control Code	7bit Encoded Control Code	4bit Encoded O code
/I/ idle	0x07	0x00	N.A
/S/ start	0xfb	Encoded by block type	N.A
/T/ terminate	0xfd		N.A
/E/ error	0xfe	0x1e	N.A
/Q/ Seq. ordered_set	0x9c	N.A	0x0
/R <sup>b</sup> / reserved0	0x1c	0x2d	N.A
reserved1	0x3c	0x33	N.A
/A/ reserved2	0x7c	0x4b	N.A
/K/ reserved3	0xbc	0x55	N.A
reserved4	0xdc	0x66	N.A
reserved5	0xf7	0x78	N.A
/Fsig/ Signal ordered_set	0x5c	N.A	0xf

표 1의 sync를 보면 데이터 기호일 경우엔 하위 2비트의 싱크 비트가 하위비트 순으로 "01"이 인코딩되고 입력받은 8레인은 인코딩 되지 않는다. 반면에 두 번째 경우 하위 2비트의 싱크비트는 하위비트 순으로 "10"이 되며 다음 64비트는 포맷을 표시하는 8비트의 타입 영역과 56비트의 인코딩된 영역이다. 입력 데이터가 컨트롤 기호(혹은 컨트롤 기호 또는 데이터 기호 혼합)일 때 가질 수 있는 경우는 총 15개이다. 인코딩된 데이터에서 빗금으로 되어있는 부분은 0으로 처리된 부분이다.

입력 데이터가 이와 다른 포맷을 형성할 경우 예러가 발생된다. 그 외에도 유효하지 않는 기타의 경우는 다음과 같다. 첫째, 인코딩된 하위 2비트의 값이 "00"이거나 "11"인 경우 둘째, 컨트롤 기호이면서 표 2에 해당하지 않는 값을 가질 경우 셋째, /O/ 코드이면서 표 2에 해당하지 않는 값을 가질 경우이다.

입력받은 8개 레인의 각각의 데이터 또는 컨트롤 기호는 각 레인을 관장하는 별도의 1비트의 컨트롤 신호가 담당한다. 총 8레인이기 때문에 8비트의 컨트롤 신호가 있고 1에 해당하는 레인은 컨트롤 기호이고 컨트롤 신호가 0에 해당하는 레인은 데이터 기호이다.

그림 3과 4는 제안하는 64B/66B 인코더/디코더의 블록 다이어그램을 나타낸다. 본 논문에서 제안하는 구조는 2단 파이프라인 구조이다. 표준 규격 동작속도인 156.25MHz의 주기 3.2ns 내에서 인코딩을 수행하기에는 전달지연 시간이 길기 때문에 인코딩에 필요한 연산 동작을 나누어 파이프라인으로 설계하였다.

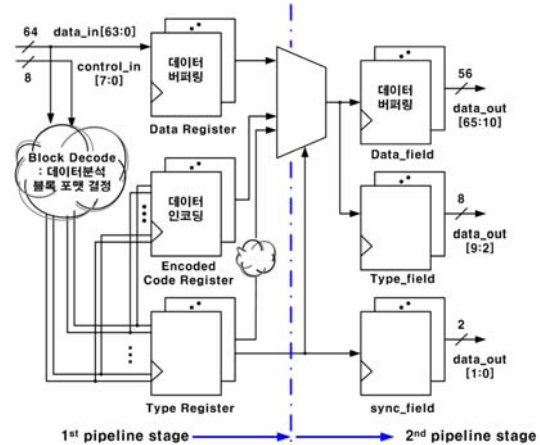


Fig. 3. Block diagram of 64B/66B encoder

그림 3. 64B/66B 인코더의 블록 다이어그램

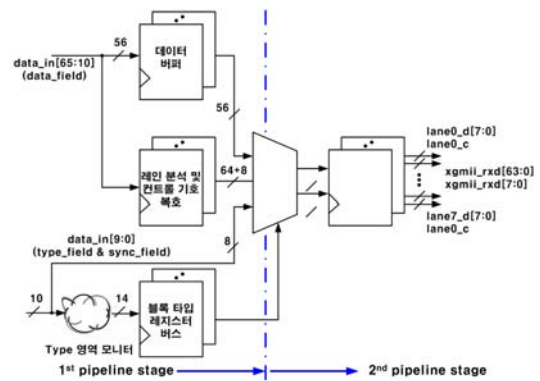


Fig. 4. Block diagram of 64B/66B decoder

그림 4. 64B/66B 디코더의 블록 다이어그램

인코더의 파이프라인 1단계에서 수행하는 주요 연산은 데이터 분석, 블록 포맷 결정 및 인코딩이다. 64비트 데이터를 8개의 레인으로 구분하여 레인별 데이터를 저장한다. 각 레인을 관장하는 컨트롤 신호와 각 레인에서 들어올 수 있는 모든 컨트롤 기호를 탐지하여 이들을 조합하여 레인 데이터 코드를 분석한다. 또한 데이터 블록 포맷을 파악하여 값을 저장한다. 위의 각 레인에서 들어오는 데이터의 기호가 파악된 모든 레인들의 내부 신호들을 연산하여 표 1과 같은 블록 타입을 결정하여 레지스터에 저장한다.

64B/66B 전송표(표 2)에 따라 각 레인의 데이터 인코딩을 하는 과정은 다음과 같다. 앞선 컨트롤 신호를 모니터링하고 기호를 분석한 내부 신호로부터 레인4의 데이터가 데이터 신호이면 저장해둔 입력받은 데이터를 그대로 출력할 것이고 컨트롤 기호 /I/, /S/, /T/, /R/ 등 인지의 정보를 받아서 그에 맞는 코드 매핑이 이루어지게 한다.

인코더의 2단계에서의 주요 연산은 인코딩된 데이터의 정렬이다. 우선 앞선 1단계에서 수행한 데이터의 블록포맷을 읽어와 이 값에 따라 미리 인코딩된 데이터 기호나 컨트롤 기호를 정렬한다. 출력되는 데이터(66b)의 레인의 개수는 입력과 같은 8개이나 각 레인 기호의 비트수는 서로 다를 수 있다.

디코더의 전체 회로도도는 그림 4와 같다. 디코더 역시 2단 파이프라인 구조로 되어있으며 디코더에서 수행하는 파이프라인 1단계의 주요 연산은 데이터 타입 분석, 싱크 헤더 분석, 데이터 디코딩, 데이터 레인의 분석은 하위 2비트의 싱크 헤더 비트값과 타입영역을 우선으로 시작한다. 싱크 헤더가 “10”일 경우 블록 데이터는 모두 데이터 기호로 되고 디코딩 작업 없이 버퍼시킨 데이터를 출력할 것이다.

디코더의 입력 데이터인 인코딩된 66비트의 블록 데이터 RXD[65:0]에서 7비트의 컨트롤 기호(/C/)가 있는 비트 위치는 표 1에서 확인할 수 있듯이 RXD[65:59], RXD[58:52], RXD[51:45], RXD[44:38], RXD[37:31], RXD[30:24], RXD[23:17], RXD[16:10]의 8개이다. 따라서 이 위치에 있는 데이터가 표 2에 의해 /I/(8'b0000\_0111), /Reserved0\*5/, /E/(8'b1111\_1110)인지를 비교하여 컨트롤 기호와 무관하게 우선 디코딩을 하고 레지스터에 저장한다. 4비트의 ordered\_set 기호(/O/)가 있는 비트 위치는 RXD[41:38]과 RXD[37:34]이므로 역시 이 위치에 있는 데이터가 /Q/(4'b0000)인지 /Fsig/(4'b1111)인지를 비교하여 디코딩을 하여 레지스터에 저장한다. 싱크 헤더가 “01”일 경우 블록 데이터는 데이터 기호와 컨트롤 기호가 혼합(혹은 모두 컨트롤 기호)된 경우이므로 해당하는 포맷을 찾아서 레인0에서 레인7까지 개별적으로 디코딩을 수행한다. 파이프라인 2단계 연산에서는 디코딩된 데이터를 정렬하는 것으로 앞선 1단계에서 읽어들인 블록 타입 영역으로 블록 포맷에 따라 디코딩된 데이터를 정렬한다.

**IV 시뮬레이션 및 측정결과**

제한한 PCS 검증을 위하여 다음과 같은 패턴의 데이터를 생성해주는 블록을 설계하고 통계 회로의 검증을 위한 PCS의 구성은 송신기와 수신기의 XSBI 인터페이스를 연결하여 송신기의 XGMII 인터페이스에 테스트 벡터를 넣어 수신기의 XGMII에서 이를 복원하는지를 확인하였다. 그림 5는 입력 TXD[31:0]의 테스트 패턴에 대한 시나리오를 나타낸다. 그림의 Lane0에서 Lane3까지는 상승 에지에서의 입력이고

	Lane7	Lane6	Lane5	Lane4	Lane3	Lane2	Lane1	Lane0
Seq0	I	I	I	I	I	I	I	I
Seq1	D	D	D	D	D	D	D	S
Seq2	D	D	D	D	D	D	D	D
Seq3	T	D	D	D	D	D	D	D
Seq4	I	I	I	I	I	I	I	I
Seq5	D	D	D	S	I	I	I	I
Seq6	I	T	D	D	D	D	D	D
Seq7	D	D	D	S	D	D	D	Q
.....	.....	.....	.....	.....	.....	.....	.....	.....

Abbreviation. D: data, S: start(0x1b), T: termination(0x1d), I: idle(0x07), Q: sequence(0x9c), R0~R5: reserved0~5, etc

Fig. 5. Test vector to verify proposed PCS  
그림 5. 제안된 PCS 검증을 위한 테스트 벡터



Fig. 6. Vector extraction by simulation  
그림 6. 시뮬레이션 벡터 추출

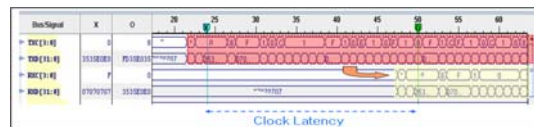


Fig. 7. Measurement using logic analyzer  
그림 7. 로직 분석기를 통한 측정

Lane4에서 Lane7까지는 하강 에지에서의 입력이다. 데이터 패턴은 컨트롤 기호와 데이터 기호를 혼합하였고 컨트롤 기호로는 초기상태기호인 /I/와 데이터 시작기호인 /S/와 데이터 종료기호 /T/를 위주로 기타의 기호를 혼합하였다. 그리고 그림 6은 이에 대한 시뮬레이션 결과를 ModelSim을 이용하여 추출해낸 것이다. 그림 7은 로직분석기 프로그램인 ChipScope를 이용하여 Xilinx사의 VertexII-1000fg456 칩의 FPGA 실장 테스트를 수행한 결과이다. 측정을 통해서 송신기의 TXC[3:0]과 TXD[31:0]이 수신기의 RXC[3:0]와 RXD[31:0]로 복원되는 것을 볼 수 있다. 파이프라인 구조를 사용한 결과 Encoder/Decoder의 레지스터는 각각 72개(Data bit:64개, control bit:8개)씩 증가되었다.

동일한 사양을 가지는 Cadence 사의 PCSR과의 성능 비교를 표 3에 나타낸다. Cadence 사의 설계된 IP 블록과 제안한 구조를 비교했을 때 제안된 구조는 사용된 F/F개수와 전체 게이트 수를 많이 줄일 수 있었다.

**V 결론**

이더넷이 발전함에 따라 PCS의 설계 공정과 구조 및 구현 방법 역시 발전하였으며 다음 세대 이더넷에 대한 연구와 논문도 꾸준히 이루어지고 있다. 본 논문에서는 8B/10B 선로방식을 사용한 이전세대 이더넷의 PCS에서 현재 10기가 이더넷의 PCS에 대한 구조 및

구현 방법을 참고하여 각 블록에 대한 구현을 수행하였다[5]-[8]. 여기서 제시한 PCS는 IEEE 802.3ae 그룹에서 표준으로 제시한 10기가비트 물리적 선로 하부계층 중 적은 오버헤드를 갖는 64B/66B 선로부호를 사용하는 10GBASE-R PCS를 설계하여 FPGA 측정을 수행하였다.

Table 3. Performance comparison of the proposed PCS and Cadence IP

표 3. 제안한 PCS와 Cadence사 IP와의 성능 비교

Performance Parameter	Cadence[9]	This work
Technology	TSMC 0.13 $\mu$ m	Magna 0.18 $\mu$ m
Number of Slices	-	1,736
Number of F/F	3558	2,141
Number of 4input LUT	-	2,167
Total Equivalent Gate Count	48,000	47,303

### 참고문헌

- [1] ETRI, "10Gbps급 이더넷 접속 칩셋 개발," 2000
- [2] R. Seifert, *Gigabit Ethernet*, Addison-Wesley, 1998
- [3] IEEE Draft P802.3ae/D1.0, "Media Access Control (MAC) Parameters, Physical Layer, and Management for 10Gb/s Operation," September 2002
- [4] IEEE 802.3ae, [http://grouper.ieee.org/groups/802/3/10G\\_study/public](http://grouper.ieee.org/groups/802/3/10G_study/public)
- [5] H. Toyoda, "100-Gb/s Physical-Layer Architecture for Next-Generation Ethernet," *IEICE TRANS. COM.*, vol.E89 - B, no.3, pp.696-703, March 2006
- [6] S. Chew, "Implementation, Verification and Synthesis of the Gigabit Ethernet 1000BASE-T Physical Coding Sublayer," *Circuits and Systems, proceedings of the 2001 IEEE Midwest Symposium*, 2001. vol.2, pp.14-17, August 2001
- [7] H. Xia, "A Mixed-signal Behavioral Level Implementation of 1000BASE-X Physical Layer for Gigabit Ethernet," *Circuits and Systems, Proceedings of the 1999 IEEE International Symposium*, vol.1, pp.431-434, May 1999
- [8] T. Yoshimura, "A 10Gbase Ethernet transceiver (LAN PHY) in a 1.8V, 0.18 $\mu$ m SOI/CMOS technology," *IEEE Custom Integrated Circuits Conference*, pp.355-358, May 2002
- [9] Cadence, 10GBase-R Physical coding Sublayer (PCSR), Technical Data Sheet, December 2006

---

 저 자 소 개
 

---

## 송 진 철 (정회원)



2006년 : 인하대학교 전자공학과  
졸업 (공학사)  
2008년 : 인하대학교 전자공학과  
(공학석사)  
2008년 1월~현재: (주)실리콘웍스  
재직중  
<주관심분야> Digital System for  
high speed interface

## 김 태 호 (학생회원)



2007년 : 인하대학교 전자공학과  
졸업 (공학사)  
2009년 : 인하대학교 일반대학원  
전자공학과 (공학석사)  
2009년 3월~현재 : 인하대학교  
전자공학과 박사과정  
<주관심분야> High Speed  
CMOS Interface IC

## 강 진 구 (정회원)



1983년 : 서울대학교 (공학사)  
1990년 : New Jersey Institute  
of Technology 전기 및 컴퓨터공  
학 (공학석사)  
1996년 : North Carolina State  
University 전기 및 컴퓨터공학  
(공학박사)

1983년 ~ 1988년 : 삼성전자(반도체)  
1996년 ~ 1997년 : 미국 INTEL Senior Design Engineer  
1997년 3월~현재 : 인하대학교 전자공학부 교수  
<주관심분야> 고속 CMOS 회로 설계, Mixed IC  
설계, PLL/DLL/CDR, High Speed Interface IC,  
Display IC